INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Université Fédérale Toulouse Midi-Pyrénées

LAAS

Département de Génie Électrique & Informatique

Research Project

Cybersecurity of industrial Control Systems 4.0

December 10, 2020

Students :

Emna	Dhouib	dhouib@etud.insa-toulouse.fr
Assa	Diarra	diarra@etud.insa-toulouse.fr
Paul Fiagnon	Etse	etse@etud.insa-toulouse.fr
Quentin	Mouret	qmouret@etud.insa-toulouse.fr
Richard	Nedu	nedu@etud.insa-toulouse.fr

Tutors :

Elodie	Chanthery
Audine	Subdias

Keywords: Industrial Control System, Attack Detection, Machine Learning, KNN, Real Time Attack Detection, SCADA, Industry 4.0

Abstract :

Nowadays, new technologies go hand in hand with computerized systems in the industrial world. In industry 4.0, systems like nuclear plants, water distribution systems etc. are completely monitored by a dedicated computer. This leads to new vulnerabilities. In a study that was conducted last year, a two-tank system benchmark was used to illustrate attacks on water distribution systems. That study used a model-based approach based on statistical inference from data to handle the attacks. However, the limitations of this method are that it is not able of handling all types of attacks and the detection is done offline. Our work aims to improve this model in order to handle any type of attack - not only known attacks - on the system, and make that detection in real time, as soon as the attack occurs. We use machine learning algorithms to build a model which can determine the normal behavior of the system so it is capable of handling any type of attack. This detection is made in real time. To make system more user-friendly, we also developed an user interface through which attacks can be simulated and executed on the two-tank model. The next step is to adapt this work to industrial needs and validate the models. Furthermore, it could be used in an existing water distribution system.

Contents

1 Introduction

This research initiation project aims to make students aware of research related activities. The project consists of two main sub-projects. Firstly, a deep literature search on a chosen subject is carried out, resulting in a state of the art. A technical part follows which consists in conducting experiments. Our group is made up of six students in computing engineering at INSA. Cybersecurity and Industry 4.0 was a natural subject choice as we all express high interest in cybersecurity. This work extends an existing project that was conducted last year by another group of students on a similar. That work used model-based diagnosis approach and residuals to identify faults and cyberattacks on industrial systems. Our present work aims to improve the aforementioned work through real-time attack detection methods through some machine learning algorithms.

In this paper, we will firstly present our state of the art. We will then describe our project management, and work attribution. A description of the benchmark is presented, followed by a brief description of the Human Machine Interface. Finally, we will present the machine learning algorithms we implemented, and the results they provided.

2 State of the Art

A Introduction

The industrial evolution has been constant since the invention of the steam machine by James Watt in 1769 [1]. During the nineteenth century, with the use of oil and electricity, the industrial production rose. This is referred to as the second industrial revolution. In the middle of the twentieth century, the use of electronic materials and automated production increased, which helped to reduce the number of workers. The industry is still evolving with the introduction of network and internet technologies. This is called the Industry 4.0. The term appeared in 2011 during the Hannover Messe [2] [3]. Cyber-Physical Systems (CPS) refers to the collaboration of computers, network systems and devices with the purpose of controlling the physical part of a system. It is used to facilitate communications of industrial devices [3].

This new model of industry has had some weaknesses since the beginning of the coexistence of electronic, network and internet technologies in the same device. Cybersecurity aspects were not always taken into account. A definition of cybersecurity will be given in the following part.

Cybersecurity is a collection of methods ensuring the protection of computer information systems from attacks to their software, hardware, electronic and network components. Its purpose is to insure the integrity, confidentiality and availability of data.

Data integrity protection refers to the ability of a security method to ensure that data is viable and accurate. Data confidentiality refers to its ability to protect data from unauthorized access/extraction. Data availability concerns its ability to ensure that the data is attainable and alterable with authorized access [4]. Cybersecurity attacks typically target one or many of these security properties. Integrity threats mostly concern malicious alteration to data; confidentiality threats involve unauthorized access to and use/theft of data and availability threats concern interruptions to data access.

This paper describes industrial systems and their link to cybersecurity threats. It provides an explanation of various cyberattacks supported by historical examples in industry. It concludes by giving a presentation of various cyberattack detection methods and their link to machine learning.

B Industry and Cybersecurity

This section provides a presentation of the an industrial system, including an explanation of the general manufacture architecture, "Supervisory Control And Data Acquisition" (SCADA) systems and a description of possible attacks on these systems.

B.1 Architecture of cyber-physical systems

B.1.1 Introduction to Industrial Control Systems and their challenges

An Industrial Control System (ICS) is the functional layer of an industrial system. Layers are composed by numerical and physical components that communicate together using wired or wireless network technologies. ICS can be found in many fields, mostly in critical infrastructures such as in energy distribution and production, manufacturing systems, transportation, defense or health [5]. ICS has been evolving since the inclusion of Information technology (IT) capabilities into existing physical systems. IT equally refers to the use of calculators, memories and network. They enabled high distance communication between connected devices of an industrial system and/or the supervision layer [6].

All critical systems are characterized by the necessity of real time communications. When a sen-

sor captures an information, it is forwarded to the top layers of the system which will react accordingly to the collected data. This industrial system can be described with the use of Computer Integrated Manufacturing (CIM) model (only real time layers are shown), which helps to reduce the complexity of ICS. As shown in Figure 1, the system is separated into five layers.

This paper will be focused on the real time portion of the system, which corresponds to the three first layers of CIM (sensor, control and supervision layers), as they are the highly vulnerable to cyberattacks[5].

The CIM model represents an industrial system as composed of layers of physical and software components. The sensors layer detects or measures physical properties. It is also referred to as the operative part (OP). It is used to collect information about the system and its environment and transmit them to the second layer: the control layer.

The control layer, also called Distributed Control System (DCS) is used to control the information delivered by the bottom layer. This command part communicates with the OP, the Human Machine Interface (HMI) and the supervision part the system. Programmable Logic Controller (PLC) can be located in command area and controls the real time system [5] [6]. The PLC has a cycle which consists of collecting data, command execution and outputs the result of its interpretation. The control layer interprets the information collected, then execute an action or generate a signal.

The top supervision layer, also called Supervisory Control And Data Acquisition (SCADA), is used for workflow and control, to produce a desired end product. It also maintains records and optimizes the production process. This layer is also used to give a view of the execution process in real time. This layer will be explained further in a dedicated section.

To link these different layers, TCP/IP protocols or analog connections are used. The first way is favored and its use is increasing as it allows rapid exchange of the data[5].

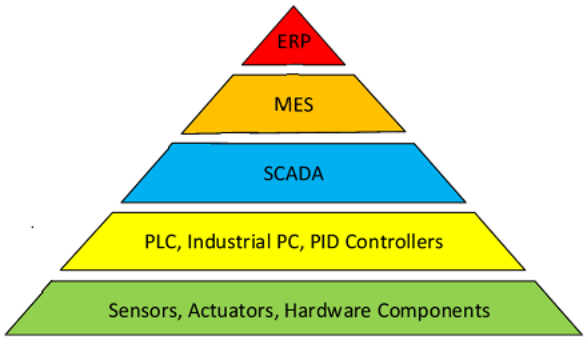


Figure 1: Automation Pyramid - Standard Computer Integrated Manufacturing (CIM)
Source: [7]

Since 2000, some solutions for ICS security have been developed for software, firmware, hardware and network industrial devices.

These solutions involve the introduction of standard or common solutions such as ISA100 or WirelessHart [5].

WirelessHart is a technology that uses encrypted data transmission collected by a sensor that transmits them to a receptor which, in turn, forwards the data to the ICS[8]. This system is based on radio transmission. ISA100 concerns wireless industrial environments, such as radio frequency for example[8]. This standard requires a physical layer based on IEEE 802.15.4, and uses IPv6 and UDP for the network layer, IPv6 is used for routing from stem to stern [8].

ISA100 is qualified by the International Society of Automation (ISA) as a "Wireless Systems for Industrial Automation: Process Control and Related Applications". They guarantee cybersecurity aspects as data integrity, confidentiality and authenticity when this protocol is used in the industrial field[8].

These solutions allow cybersecurity characteristics of the IT field and are in actuality hardly applicable in the industrial field, more precisely in a real time context[5]. Most of the solutions proposed by the IT field are not sufficient for ICS systems because of the real time constraints. On top

of that, the heterogeneity of industrial systems and the 24h/7 availability of the factory makes it more difficult.

- For instance, these constraints do not allow the use cryptography. The other difficulty in the industry, is the possibility to have a breakdown which can be interpreted as an attack and the other way around.

As seen before, ICS have a lot of vulnerabilities and different types of attacks can be performed on each layer[5].

To give an idea of attacks that can be executed on the different layers, some of them will be introduced in this part and developed in a next section.

The first and second layers are vulnerable to data injection and modification of the information given to the sensors. The system will then react considering these false information, which is not a reaction that is naturally adapted to the real situation (e.g. the Stuxnet attack in Iran in 2010 ref: B.2.2 and the Maroochy Shire attack in Autralia in 2000 ref:B.2.2) [5].

The SCADA layer is mostly vulnerable to Distributed Denial of Service (DDOS), replay B.1.2 and Man In The Middle (MITM) attacks. During a MITM attack, the attacker is placed in the path of communication between the source of the information and the receiving machine which allows him to intercept, modify or spoof the data[4].

B.1.2 Description of "Supervisory Control and Data Acquisition systems" and their vulnerabilities

The term SCADA or "Supervisory Control and Data Acquisition" systems can be used to describe the entire ICS architecture[9]. They are commonly used in industrial installations as they enable efficient collection and analysis of data, provide manageability and maximum reliable control of industrial equipment such as sensors and controllers over long periods of time. The industrial equipment are small computers using standard computer elements such as embedded operating systems, communication protocols, accounts,etc.

However, SCADA systems were not designed to be resilient to cyberattacks as they are usually located in remote and not easily accessible areas, and use special communication protocols [10]. These systems were actually designed when Internet access was not known to be a risk [11]. Hence, they are nonetheless vulnerable to traditional cyberattacks.

Advances of networks, the internet, networking communication protocols and the critical nature of the SCADA systems made them an interesting target for hackers. As a matter of fact, infrastructure systems can be attacked from anywhere worldwide which can have dramatic tangible consequences [12].

These systems are mostly targeted by Denial of Service (DOS) and code injection attacks [9]. A DOS attack is cyberattack that consists in making a system inaccessible, prevent communication between its components or disturb its activity by preventing it to provide its intended service. A variant of this attack, called Distributed Denial of Service, consists in attacking the same system from multiple sources at once[4].

Code injection is a cyberattack that consists in adding a segment of code to an initial code. It commonly refers to the communication of malicious information to a system. A variant of this attack, called replay attack, consists in recording data from system equipment (controllers and sensors) and replacing the actual data with the recorded one in order to deceive the program[13].

These attacks target data availability and integrity, respectively. They are usually carried out throughout the exploitation of an alternative entry access to the system. This access can be a remote access port used for maintenance, a channel between regular IT systems and SCADA systems, a malicious link or removable devices. This provides the hacker with the means to control and send commands to the devices and interfere with their activity[10].

B.2 Description of possible attacks on industrial systems

B.2.1 Presentation of cyberattacks on industrial systems and their motives

A system can be attacked for various motives, with different intents. However, the type of attackers and their reasons can be summarized into four main categories in the following section.

The first one is the gathering of intelligence on physical operations or intellectual property. It can be for military purposes, such as in the "Stuxnet" attack B.2.2.

Another motive is hacktivism. In this case scenario, an attack is performed to change the ideas and motives of an industrial group. The hacker can steal confidential information and divulge them or perform a DDS attack (e.g. Dyn Managed DNS attack 2016 [14]).

One could also attack a system with criminal intents. The primary purpose of this type of attack is financially-based. It generally does not damage machines. Instead, data and/or access means to the system are gathered and sold.

The last possible attacker is the individual, he is performing the attack alone, for its own reasons or motivations who can be politic, revenge or personal. It can be everyone even an employee, who can have some privileged access upstream. He may have access to confidential data or may have had access in the past, (e.g. Shamoon attack on Saudi Aramco 2006 was performed by an individual attacker ref: B.2.2) [15] [11].

As shown before, IT technologies are increasingly used to ease performance in the industry. As previously described, attacks on Industry 4.0 systems lead to consequential loss in human life and economical sector. During the last decade, the dreadful consequences of these attacks raised the serious concern of protection methods of these systems. In the following lines, some important historical attacks performed on ICS or computers used for non critical operations) [11] will be presented.

An Attack can be performed on ICS's layers or on the internal computer network. These attacks can also be dangerous for ICS if they are connected. ICS are mostly vulnerable to attacks such as : phishing, viruses, customized malware, insider complicity, worm, spyware, spear-phishing, backdoor, Trojan [11] [15] [5].

Phishing consists in getting access to confidential data using fraudulent way such as the impersonation of as an authorized figure.

A virus is a code segment that infects other parts of a system and impacts the confidentiality, integrity and availability of data, once it is executed (e.g. Shamoon attack ref: B.2.2) [4].

A worm is an autonomous program, that can spread on the system and infect all the machines (e.g. Stuxnet attack ref:B.2.2, Duqu attack ref:B.2.2).

The purpose of a spyware is to infect machines to collect information (e.g. Flame attack ref:B.2.2).

A Trojan or backdoor attack consists in creating a breach in the system that allows the attacker to have a complete access to the system and exploit it freely [16] ref:B.2.2.

B.2.2 Description of the attacks

- Shamoon: this attack targeted the energy sector, more precisely the gas and oil industries. It destroyed data on the infected machines. This virus was introduced in the system by an insider accomplice using an USB stick [15] [11].

- Stuxnet: targeted attack on an Iranian ICS system in 2010. It destroyed centrifuges by forcing them to spin at a higher speed than they were supposed to.

- Duqu: used for gathering information from ICS, was discovered in 2011 [11].

- Flame: transmitted via USB stick and discovered in 2012, it collected and forwarded

a maximum information from infected computers to the remote attacker[11].

- The first consequential attack on an industrial traffic communications control system took place at the trans-Siberian pipeline in 1982 and lead to an explosion in space [17]. A Trojan had been planted in the system that controlled the pipelines and caused an explosion of 3 kilo-tone of TNT (trinitrotoluene) that was visible from space.

- Chevron is an American multinational energy corporation based in New York. In 1992, one of its old fired employee hack the firm's computers and disabled its alert systems. He then reconfigured the computers to crash. This attack has been discovered ten hour after an emergency arose at the Chevron refineries. It put thousands of people at great risk [12].

- Salt River project is a corporation that work on water distribution and electricity utilities in the state of Arizona, United states. In July 1994, an attacker hacked into the firm's computers' network and installed a back door that gave him access to the system. The system was used in water distribution in the Phoenix metropolitan area. This access gave him control over the canals and access to important information such as financial, customs data, root passwords, system logins, etc. [12].

- The control system of air traffic communications of Worcester's (Massachusetts) Airport was disabled by an attacker that penetrated the system. He blocked many services like the Federal Aviation Administration control tower, airport security service, and the weather service for six hours [12].

- Gazprom is a Russian company that works on extraction, production, transport, and sale of natural gas. An attacker collaborated with a company employee in 1999 to access the switchboard of the control central. He then controlled the gas flow in the pipelines.

- The California Independent System Operator is a non-profit Operator that manage the state of California's electric power system, transmission lines and electricity distribution. In 2001, attackers from china took the control of Independent System Operator's computer network in California for more than two weeks [12].

- Maroochy Shire attack was achieved in Australia in 2000 on an industrial sewage spill, the supervision/ control part of this industry was communicating with the pumping station via radio frequency. The attack lead the system to increase the number of fault. The result of this attack performed on SCADA is the release of 800 000L of sewage[5] [18].

In recent years, there has been many others attacks that has not been presented above. These attacks have heavy consequences in terms of financial loss and human safety. Thus, the Industrial Control Systems need to be protected. That protection must start with attack detection which will presented in the next section.

C Detection Methods

C.1 General methods of detection

Intrusion detection systems (IDS) [6] are tools designed to detect malicious activities on the target they are monitoring. They are used in addition to traditional solutions such as firewalls and trigger an alert when malicious behaviour is detected.

A malicious activity can be an attempted unauthorized access, an illegal access, the modification of data, or the decommissioning of one or more computer components.

Identification can be done in real time on the network to immediately detect threats or by analyzing previously captured network traffic.

These detection be classified into two categories, broken down by detection method. The first category is based on regularly updated signatures. The other one is a detection system based on the identification of abnormal network traffic.

C.1.1 Signature intrusion detection systems

Signature Intrusion Detection Systems (SIDS) [19] [5] are based on attack description libraries (signatures). During network flow analysis, an alert is issued when a signature is detected within a packet. This detection methodology proves to be effective only if the signature base is kept up to date on a regular basis. In this case, the detection by signatures is highly reliable.

Several implementations exist to perform signature detection, such as decision trees or state transition systems.

C.1.2 Fault Intrusion Detection Systems

Dissimilar to SIDS, the Anomaly-based Intrusion Detection System (AIDS) [19] does not rely on attack description libraries. They are in charge of detecting abnormal behaviours when analyzing the network flow. The system is then based on two phases:

- A learning phase, during which the system studies the network flow behaviour.
- A detection phase during which the system identifies abnormal events.

This intrusion detection technique is recognized as highly effective.

C.2 Machine learning

C.2.1 Introduction to machine learning

Machine learning is one of the most commonly used applications of artificial intelligence (AI). It provides computers with the ability to learn automatically after providing them a large amount of data. Nowadays this application has become a widespread technique worldwide with possible applications in different fields. It intervenes in various fields, in particular medicine, robotics, education, military and cybersecurity. By developing the computer algorithms, machine learning has the goal to predict the behaviour or state of an unknown situation by exploiting big data inputs[20].

There are two categories of machine learning algorithms[20]:

- Supervised learning:

  It consists in finding a rule to predict the outputs. The inputs are, in consequence, labeled, which needs a consequential amount of manual work.

- Unsupervised learning:

  It aims to explore the inner characteristics of the input data, which does not require any label on the inputs.

These two categories of algorithms are complementary. The supervised learning helps with the pattern classification and data mining which is used by the unsupervised learning algorithm. Deep-learning is a part of machine learning [21] [2] that allows to extract big data and handle it from sensors, which makes it an essential utensil in industries.

C.2.2 The use of machine learning in attack detection

In recent years, techniques of machine learning is being used to analyze big data to extract relevant information. It can analyze data, make classification groups and help in predictions.

Machine learning is being increasingly used to detect and classify malware. Attacks evolve every day; a technique of detection is specific to one or a certain type of attack. Attackers are progressively improving their techniques. Thus, techniques of

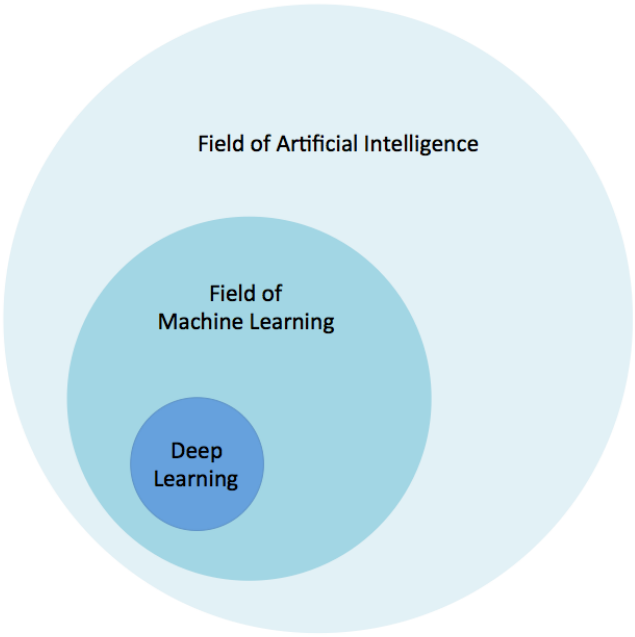


Figure 2: The relationship between AI and deep learning

Source: [21]

detection need to be adapted to handle these new forms of attacks. This shows the necessity of machine learning in detection methods as models can adjust their behaviour to detect new forms of attacks [22]. Models can also learn from new attacks on their own and adjust their behaviour in result. In the following lines, some real cases will be presented, where machine learning has been used to detect attacks.

Daniel Gibert et al [22]. present two methods based on machine learning for malware detection in computers. Their approach consists in an analysis of the software. Their first method approach, called "static method" is to analyze the source code of the software. This analysis is based on sequences of characters present in the code, system function invocations or the occurring of some bytes in the source code. They also present an approach of representing the source code as a gray scale image. These elements of the source code are extracted to create a dataset.

The dynamic method is based on the behaviour of a program during execution. The network traffic, system calls, information retrieved from memory usage are analyzed and used to build datasets. The models created from these datasets can then detect malware when a real program is running.

Version	**IHL**	**ToS**	**length**	
Identification			**Flag**	**offset**
Time to live	**Protocol**		**checksum**	
Source address				
Destination address				
Options				**Padding**

Table 1: IPv4 packet structure

Philip K. Chan et al. [19] present a new approach to the use of machine learning in attack detection. Contrary to commonly used methods that detect a specific attack on a system, they propose a method that is built on the normal behaviour of the system as to detect unknown threats.

The models are built on the desired operation of the system. Hence, when the current operation is not considered normal by the models, they designate it as an attack. The disadvantage of their methods is that the will handle every unknown behaviour as attack despite the fact that not all of them are attacks. This methods can be used on systems that require a lot of safety.

Tao Xia and al [23] work on how to detected intrusion attack in the network. They proposed a based on information theory and generic algorithm to detect network intrusion. Their method consists of analysing the packet in network and detect if it is vulnerable or not.

They used information theory of Shannon to create their dataset. A network packet has many fields and each field stock a certain type of information as shown in table 1.

Each information is linked to probability. Thus, fields that are not relevant get low probability and the important fields get high probability. Some fields of the packet for like version and checksum are determinant to the classification of packet as

vulnerable. The important field that they considered in their dataset is the service type, the protocol type, some flags.

After building their datasets, They used generic algorithm to build their models. They test their models with 972781 normal data and got 16132 false positive detection. False positive is a result where the model detect an intrusion but normally it must not. The rate of false positive detection is 1.66%. They test their model with 3925650 abnormal data and got 29558 of false negative detection. A result is considered as false negative when a model does not detect an attack where it must detect normally. Hence, the rate of false negative detection is 0.75%. These rates are acceptable. As a result, their models can be considered as correct.

These mechanisms can be regrouped into two categories, depending on their approach: data-based (supervised learning), or model-based (unsupervised learning) [23] [24].

C.2.3 Unsupervised Approaches

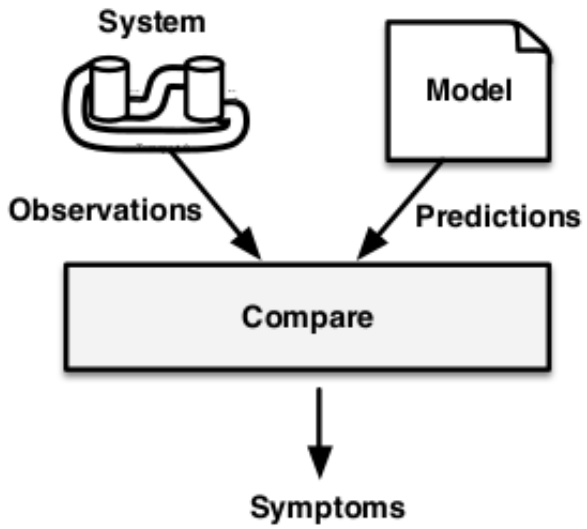


Figure 3: Model-Based Approach to Anomaly Detection

The unsupervised approach is divided into two steps. Firstly, the Model-Based approach is used to detect symptoms, implying detection of anomalies and potential attacks. Figure 2 [24] explains that mechanism. The model is based on all the data the systems provides. As example, the sensors of a car are modeled using characteristic maps, and a chemical process can be described by differential equations. If a reactor is installed, the chemical and physical process is modeled, e.g. using differential equations. When a sensor sends data different from the data expected by the model, the comparison is wrong. The user can then get the symptom and try to use it to find the root.

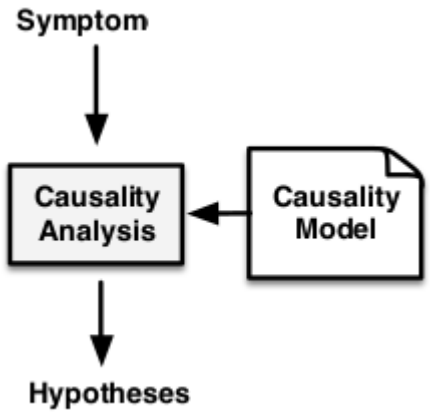


Figure 4: Model-Based Approach to Diagnosis

Once the symptom has been identified, the model-based approach is used to figure out the root of the anomaly. As previously, and as explained in figure 3 [24] symptoms are compared with causality models, and from this comparison results the most likely hypotheses of the root of the anomaly.

The main drawbacks of this approach is that a system can have a great number of different causes, and consequently as many models, which are handmade tools. It also difficult to obtain verified models.

C.2.4 Supervised Approaches

In a detection system using a model-based approach, the training data will have a known re-

sult that the system can use to refine its detection model. A system with a data-based approach cannot have this feature, as the data used has no indication for the output to be achieved.

An intrusion detection system implementing this approach has interesting properties.

Foremost, there is the reliable detection of outliers coming out of the data stream. This feature is fundamental for detecting anomalies in a network.

Furthermore, there is little interaction during the learning phase, which allows to manipulate huge amounts of network data that were not previously labeled as normal or abnormal.

At last, in the event of a change in the network topology or any other peculiar change, the mechanism based on unsupervised learning will gradually adapt. This phenomenon is called adaptation to change.

C.2.5 Data clustering

Data clustering [19] is a method of data analysis that allows a set of data to be divided into groups with common characteristics. This method enables unsupervised learning solutions to be implemented as to automatically deduce data structures by pooling similar data. Once partitioning is done, some outliers may not belong to any group, as they differs significantly with the other data present in the set. In the context of network intrusion detection systems based on abnormal behaviour, outliers can be used to identify abnormal traffic in a computer network.

Several algorithms can be used to implement a clustering network intrusion detection solution. Here is a list of algorithms commonly[25] used in this field:

- the K-means algorithm, used to find K clusters in a circular way.
- the DBSCAN algorithm which is used to find a previously unknown number of clusters. This algorithm works with any type of "shape" the data set.
- algorithms belonging to the hierarchical grouping which is used to find a previously unknown number of clusters according in a circular way.

D Conclusion

Industrial Control Systems are an architecture adopted to manage big infrastructures such as nuclear plants or water distribution systems. All the information are collected by sensors and machines transmitted via the network to the controls center. In response, the control center sends instructions to the physical system.

The vulnerabilities in these systems come from the heterogeneous interactions between the different components and technologies in the industry 4.0. These systems were created without taking into account cybersecurity aspects. Technical methods applicable in IT field can not be used in industry.

This paper has introduced some examples of attacks preformed on these systems. It exposes the significant number of vulnerabilities of an industrial system 4.0, describes techniques used to detect attacks and presents some possible solutions to avoid, resist or resolve attacks.

The principle of detection techniques are to model tools that understand the normal operation of the system and detect attacks when an abnormal behaviour or event occurs. A machine has the aptitude of deducting relevant information from large sets of data. The use of machine learning helps to improve methods of detection.

3 Project Management

This project was conducted in the context of an initiation to research and was supervised by two tutors: Audine Subias and Elodie Chanthery. As stated in the introduction, our work extends a study that was carried out last year by a different group of students. In this part of the report, we will expose the tasks and responsibilities of each member of the group. Then we will provide a brief overview of the scheduling of the project. Finally, we will present the design techniques and software used throughout the project.

3.1 Project Scheduling

The entirety of our project was carried out by the Agile Methodology. That methodology consists in setting short-term objectives by dividing a project into several sub-projects. The need was outlined by the tutors, after which we produced a first list of objectives to be achieved by a given deadline. This allowed us to divide the project into several tasks. The group was then divided into small sub-groups. Each task was carried out in order of priority by designated sub-group. This technique allowed us to focus on the tasks that were important to our tutors. When a task is completed, demonstration is carried out then validating. We made a provisional planning with the main tasks of the project 5.

We planned a meeting every fortnight to discuss the evolution of the project and the compatibility of the achievements with the expectations of our tutors. This was an important step to reestablish the requirements of the project progressively.

In general, before a working session each group has to give a brief summary of the evolution of their assigned tasks. Before a meeting with our tutors, we made a summary of the evolution of each assignment in order to follow the progress of each work and prepare questions to better target needs or to ask for further directions.

3.2 Tasks, Roles and Responsibilities

The first goal of each part of the group was to discover the script on MatLab and to test the simulator with different scenarios through the Human Machine Interface. The purpose was to help us familiarize with a new environment more related to electronics than computer science. These sessions also provided an opportunity to discover the different aspects of the project as a group and to meet with our tutors to define the objectives to be attained. We then carried out tasks assignment to the different sub-groups and an in-depth study of the of each task.

In order to achieve our main goals and learn more in depth the different aspects of the project, the work was separated in three main parts.

- **Human Machine Interface (HMI) group**
  A HMI was designed by the previous group which allowed a user to choose a fault or attack scenario. For our work, we decided to create an interface that allows the user to change each parameter of the simulation model. The parameters are modified to imitate attacks or fault. Throughout the sessions, the objectives evolved as follows :
  - the design of the interface
  - the scripting of the interface on Matlab Apps
  - the insertion of a 3D animation of the two tanks on the interface
  - the reflection on the parameters that could be modifiable from the interface
  - connect the interface to the model on Simulink
- **Machine Learning (ML) group**
  The work that was carried out last year only allowed the detection of abnormal behaviour after an attack was executed. The goal of our study is to find a way to detect attacks and

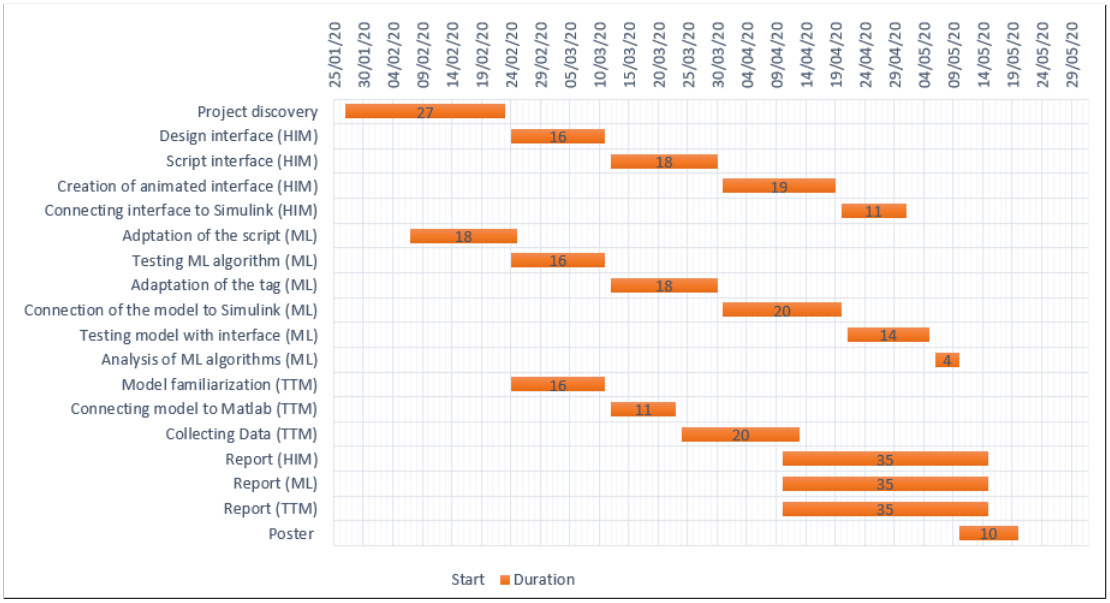


Figure 5: Gantt Diagram for the provisional planing

classify them as they occur. Throughout the sessions, the objectives for this part of the project evolved as follows :

- modify the script in order to adapt tags and test ML algorithm on Matlab with the available data
- on each second of the simulation, each set of variables have to be tagged as normal or error
- link the detection model to Simulink
- test the efficiency of the detection model by connecting the interface of scenarios with the model
- test various machine learning algorithms on available data and analyse the results

- **Two-Tank model (TTM) group**
  The first part of the project was carried out on another model. This year the goal was to discover the new two-tank model, to connect it to Simulink and to collect a quantity of data that would eventually be used in the ML part. Our objectives evolved as follows:
  - familiarize the group with how the model works and runs simulation
  - connect the model to Matlab and replace the Simulink model by the real model to collect data
  - with the containment the objectives of this part evaluate and the purpose was to create a 3D simulation with Simulink

The first idea was to work in groups and move on to a different part of the project every two or three sessions. Due to the COVID-19 lockdown, we did not have access to the actual benchmark model. We also had to give up on some goals as we lacked the resources necessary to achieve them. We notably had to renounce inserting the 3D animation into the HMI. Our previsional planning 5 evolved considerably compared to the planning we first had establish6.

3.3 Design Technique

To better highlight the interactions between the different component of the project (i.e.HMI, ML model, Simulink), we designed Unified Modeling Language (UML) diagrams. This also helps

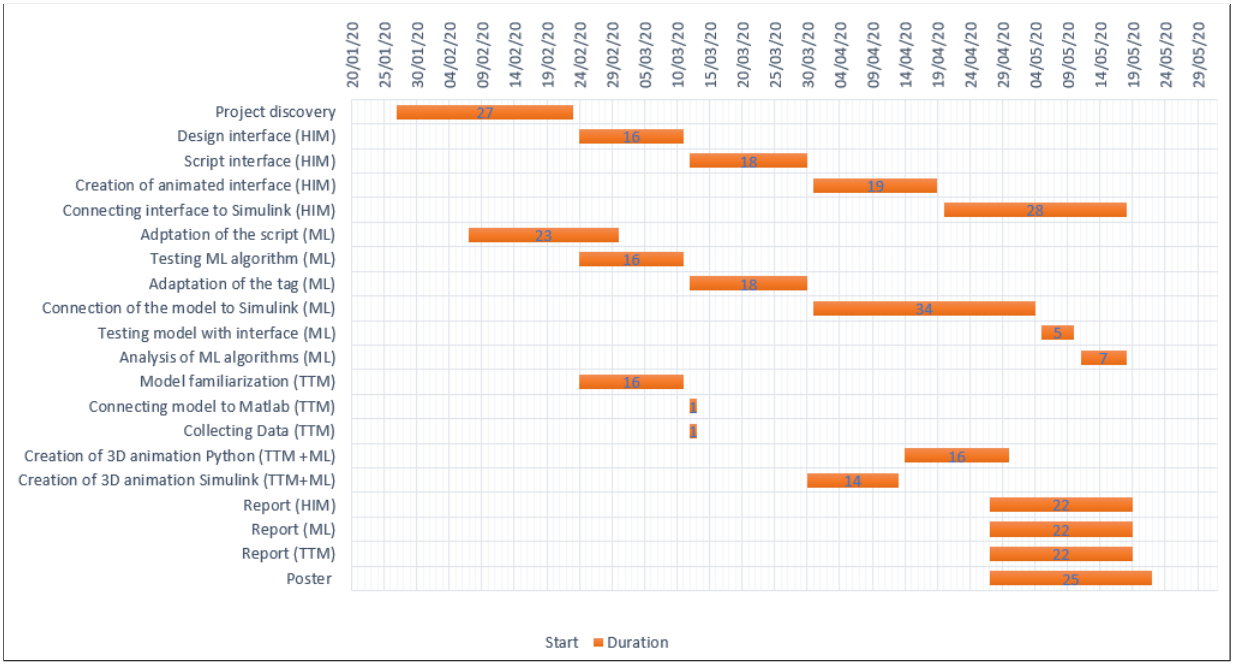


Figure 6: Gantt Diagram for the real planing

establish better communication between group members.

3.4 Software used in the project

In this project the main software used are Simulink and Matlab. Simulink was used to run simulation on the two-tank model, it is the graphical programming part. Matlab was used to create the graphical interface and to testing various machine learning detection algorithms.

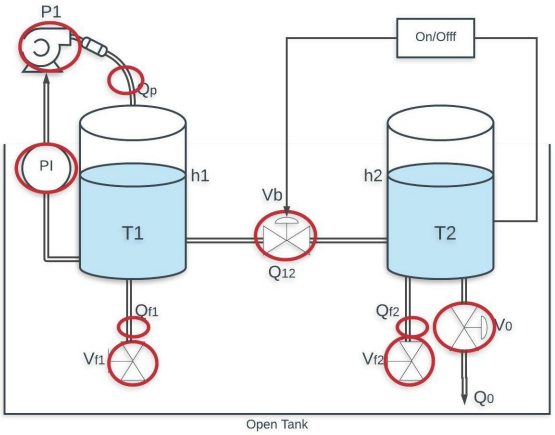


Figure 7: Two-tank System

4 Benchmark Description

7 shows a representation of the benchmark. It is a remodeling of the two-tank benchmark used in [26]. They both aims to provide water to customers in a continuous flow Q_0 through valve V_0. It is a two tank system T_1 and T_2 connected in series by a control valve V_b. Our system uses the same variable names as in [26]. The X^m sign is used to indicate that variable X is being measured.

Tank T_1 is filled by pump P_1, with a measured output labeled Q_p^m. P_1 is controlled by PI controller with a measured output of U_p^m. Each tank can signal a leak flow noted V_{f1} for tank T_1 and V_{f2} for tank T_2. Sensors are used to calculate the water level h_1^m and h_2^m of tank T_1 and T_2 and can be affected by a leak or an overflow or a simulation of those. As stated in [26] the execution of the faultless mode is described in details in [27].

The input of the system can be modified through an HMI. It is then possible to act on the different parameters of the system (circled in red in the figure 7) such as the leak rate in each tank through V_{f1} and V_{f2} or the overflow rate of tank T_1, named Q_p^m, from pump P_1. The position of the valves V_0 and V_b, respectively through the variables U_0^m and V_b^m, can be modified to stuck them open or close for example. In the same way, the value of U_p^m, which corresponds to the PI controller output. The PI controller can be turned on or off thus affecting U_p^m. The same goes for the sensors U_p, Q_p, P_1 and P_2 with their respective outputs U_p^m, Q_p^m, P_1^m and P_2^m. We consider that a parameter's default value (resp. faulty value) is 1 (resp. 0).

5 Editable Interface for attack injection

As previously discussed, a HMI was designed to allow an "attacker" to modify the input values of the benchmark model. This interface makes fault and attack simulations more flexible by directly launching simulations on different parameters of the system. The interface is inspired by the HMI implemented in the previous work that this study extends. That interface simply considered different predetermined faults and attack scenarios.

The new HMI was designed through the Matlab Apps tools. It has different input parameters connected to the Simulink Model. This Simulink model is based upon the previous model which has been modified to allow direct changes to specific subsystems of the benchmark.

For example, is possible to change numeric values from the interface such as leak and overflow rates (as described in f5) 3 or switch valves open or close and controllers on and off. These changes are sent to the Simulink model in real time during simulations. Multiple scenarios can be adapted to different time slots or combined with other scenarios.

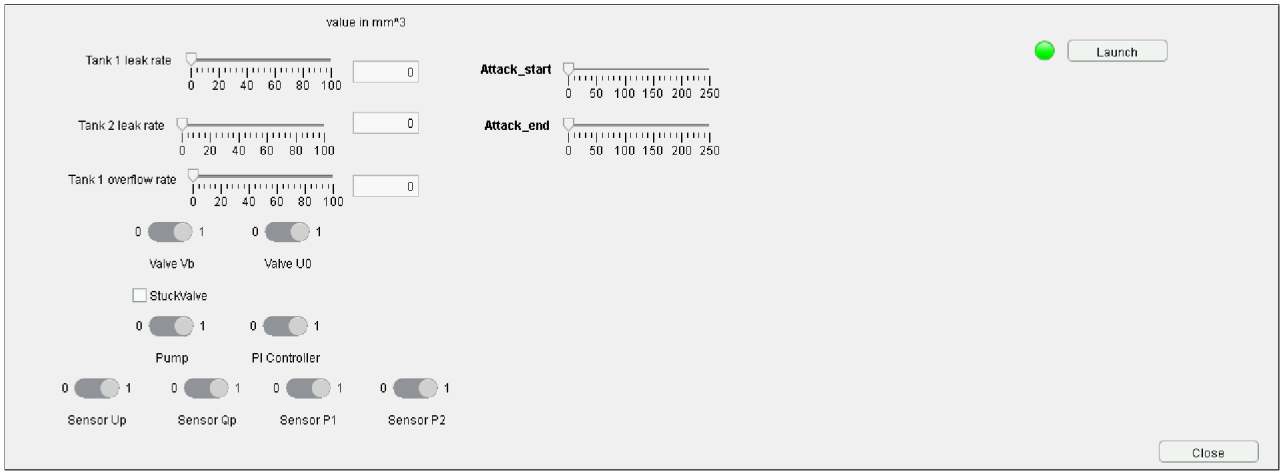


Figure 8: Preview of the HMI

6 Machine Learning detection

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. ML algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Here we use ML techniques to detect attack on the benchmark.

ML algorithms can be divided into two groups : Supervised and Unsupervised Learning approach. Supervised learning is a technique that consists in accomplishing a task by providing training, input and output patterns to the systems whereas unsupervised learning is a self-learning technique in which the system has to discover the features of the input data on its own with no prior set of categories used. Here we use the supervised learning approach. The data was tagged before training.

Matlab [28] offers a technology known as Classification learner [29] which easily allows us to build ML models in a few clicks. It also gives statistics such as accuracy and the convolution matrix about the model. We used this app to build our ML models.

In the following lines, we will first present an overview of our ML Block. Then we will demonstrate how data has been collected for the training of the models. And Finally, we will present how we use those models in our system and the results that we obtained.

6.1 Overview of the ML Block

Our goal here is to use ML techniques to analyse the behaviour of the system and detect if there is a problem or not. Here we come across a classification issue because our outputs are not in numeric values. We use the following features are :

- the level of water in tank 1
- the level of water in tank 2
- the debit of water from command b
- the debit of water form command p
- the state of valve b (close or open)
- the state of valve o (close or open)

The output of our system either indicates an attack or an error. The prediction model predicts the state of our system from the six features shown in the figure 9. In order to build our model, we collected data from the behaviour of the system in known conditions. We then created different detection models with the collected data to use them in our benchmark.

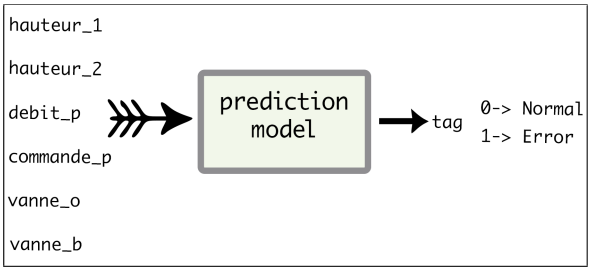


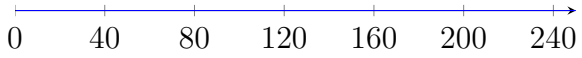
Figure 9: Overview of ML Block

6.2 Training

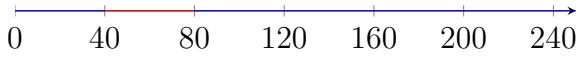
In order to build our models, we collect data from the behaviour of the system. We use a benchmark from a previous system [26]. That system presents a set of 27 scenarios of execution of our benchmark. Each scenario describes how the benchmark is supposed to work during its run. Some of them simulate the disfunctionment of specific parts of the benchmark while others simulate an attack. The table 2 shows all our scenarios and their descriptions. From these scenarios we know the exact tag of the system : normal or error.

As we can see the scenarios from the table can be classified in four groups:

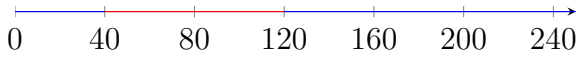
- A faultless execution where there is no attack from the beginning to the end of the run time.



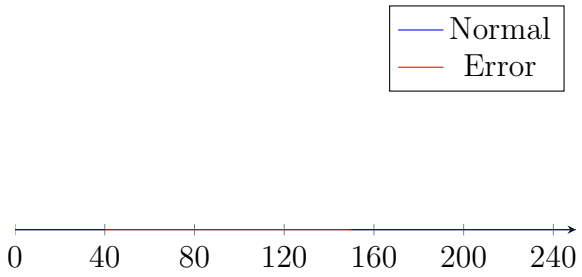
- An execution where an attack is being performed on a set of data from 40 to 80 seconds. Outside of this time slot, the system runs normally.



- An execution where an attack is being performed on a set of data from from the 40^{th} to 120^{th} seconds



- An execution where an attack is being performed on a set of data from from the 40^{th} to 150^{th} seconds.



The table 3 presents for each scenario, the attack group it belongs to. We run ten simulations by scenario. We obtained as a result a data set of $26 * 250 * 10 = 65000$ inputs.

We first inserted the information collected during the training in our data set. Then we built our prediction models in the Matlab Classification Learner App [29]. After building the models, we exported them and saved them in files by using the `saveCompactModel` function of Matlab. We tested our data with a few set of algorithms which's results are presented in the section 6.4. The use of the model in Simulink [30] will be presented in the next section.

group of attack	**scenario**
faultless	1
attacks from 40 to 120	2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 16, 17, 20, 21, 23, 24, 25, 26, 27
attacks from 40 to 80	14, 15, 18, 19
attacks from 40 to 150	8, 12

Table 3: Groups of scenarios

6.3 Usage of ML models

Here we are going to present how our ML models has been used in the benchmark. The benchmark is a model that has been designed through the Simulink app. During simulation data is transmitted through signals in the app. Firstly, we locate and redirect all the outputs of the benchmark to a single port. This port puts out at every second of the simulation the features of our ML model : hauteur_1, hauteur_2, debit_p, commande_p, vanne_o and vanne_b. Then, we use a Matlab Function block in Simulink that is connected to the outputs of benchmark. This block imports the model from files which remains persistent in a Simulink Function block. During the simulation, that block calls the prediction function at input signal and gets the corresponding tag of the system.

Listing 1: KNN prediction bloc

```
function y = fcn(u)
% u is signal that
%contains all the features
persistent mdl;
if isempty(mdl)
    mdl = loadCompactModel('
        KNN_model');
end
     pred = predict(mdl,u);
y=pred;
```

The output of that function is displayed in a scope.

6.4 Comparison the ML algorithms used

We tested out data with many ML algorithms. Here we present the principle of each algorithm used and their results (table 6.4).

- KNN (k-nearest neighbors) is a ML algorithm used in classification and regression problems. It stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN is one of the algorithms that provides the best results on our system. Its matrix of convolution is presented in figure 10.

- Ensemble classifier algorithm combines several machine leaning algorithms to build a new classifier that may differ in the algorithm used, hyper-parameters, representation or training set.

- Decision tree are a non-parametric supervised learning method used for classification and regression. It used data to learn and approximate a sine curve with a set of if-then-else decision rules.

- Discriminant is a dimensionality reduction technique which is commonly used for supervised classification problems. It is used for modeling differences in groups (i.e. separating two or more classes). It is used to project the features from a higher dimension space into a lower dimension space.

The table 4 presents the result of each algorithm.

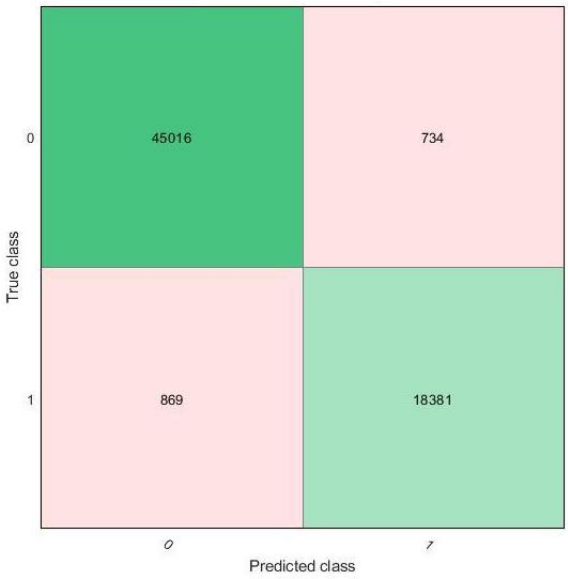


Figure 10: KNN confusion matrix

	Algorithm	**Accuracy**
1	KNN	97.5
2	Ensemble classifier	97.5
3	Tree	90.7
4	Linear discriminant	71.8

Table 4: Accuracy of algorithms used

The accuracy is the ratio of number of correct predictions to the total number of input samples.

	Fault/Attack	start	end
1	faultless mode	-	-
2	pomp fault	40	120
3	level sensor fault of tank T1	40	120
4	level sensor fault of tank T2	40	120
5	tank T1 leak fault	40	120
6	tank T2 leak fault	40	120
7	sensor fault mUp	40	120
8	valve Vb stuck closed	40	150
9	valve sensor fault mUb	40	120
10	sensor fault Qp	40	120
11	PI controller KO	40	120
12	valve Vb stuck open	40	150
13	sensor T1 default # 0	40	120
14	short term water theft of T1	40	80
15	short term water theft of T1 without any change of my1	40	80
16	long term water theft of T1 without any change of my1	40	120
17	long term water theft of T1 with a small change of my1	40	120
18	short term water theft of T2	40	80
19	short term water theft of T2 without any change of my2	40	80
20	long term water theft of T2 without any change of my2	40	120
21	long term water theft of T2 with a small change of my2	40	120
22	replay attack (not used)	160	200
23	long term T1 overflow with simulated leak	40	120
24	long term T1 overlow with TI pass by	40	120
25	Empting T2 without leak	40	120
26	Theft in T2 when user is not pumping water	40	120
27	Teft in T2 when user is pumping water	40	120

Table 2: Attack scenarios

7 Conclusion

This project builds upon a study from last year students. It has been redesigned to improve the safety of the system. The HMI was modified to allow attacks and faults simulations through parameters instead of fixed scenario. Machine learning is used to identify the state of the system by analysing it is behaviour. A data set is built by collecting information on how the system works then used to create machine learning models through Matlab Classification Learner app. These models are used in the system to detect different attacks. Our most proficient models give accuracy up to 97.5%. The present work not only shows better results than the previous study, it also provides a way to detect attacks as soon as they occur. Most of our study has been done through different simulations. This work could be used in the future on a physical model.

Throughout this project, we have developed different project management skills such as organization, coordination and progress monitoring, time management, communication and collaboration between the team members. The unexpected COVID-19 lockdown was both an opportunity as to make our working hours more flexible, but also a challenge on the organizational level. We had to be rigorous and to set frequent milestones in order to progress.

References

[1] Insee, "Histoire industrielle," *Insee Nord-Pas-de-Calais*, vol. 14, 2017. [Online]. Available: https://www.insee.fr/fr/statistiques/2121532

[2] S. Pfeiffer, "The vision of industrie 4.0 in the makinga case of future told, tamed, and traded," *Springer.com*, vol. 15, 2017. [Online]. Available: https://rdcu.be/b073U

[3] N. Jazdi, "Cyber physical systems in the context of industry 4.0," *IEEE*, vol. 3, 2014. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6857843

[4] E. Alata and V. Nicomette, "Security of tcp/ip networks : introduction," pp. 2–5, 2019. [Online]. Available: http://homepages.laas.fr/nicomett/SECU_RESEAU/slides.pdf

[5] J.-M. F. Franck SICARD, ric ZAMAI, "Cyberdfense des systmes de contrle-commande industriels : une approche par filtres base sur la distance aux tats critiques pour la scurisation face aux cyberattaques." *HAL archives ouvertes*, vol. 17, 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01654260/document

[6] K. Stouffer, S. L. Victoria, P. Marshall, and A. Hahn, "Guide to industrial control systems security," *National Institute of Standars and technologies*, vol. 255, 2014. [Online]. Available: http://www.gocs.com.de/pages/fachberichte/archiv/164-sp800_82_r2_draft.pdf

[7] P. Kadera, "Methods for development ofindustrial multi-agentsystems," *DOCTORAL THESIS: University in PragueFaculty of Electrical Engineering*, 2015. [Online]. Available: https://pdfs.semanticscholar.org/6b45/b416408d48a5758e54a34161012b39716175.pdf?_ga=2.115044487.1615576853.1580670203-1962436537.1579776723

[8] M. Nixon, "A comparison of wireless hart and isa100.11a," *Emerson Process Management*, vol. 32, pp. 1–14, 2012. [Online]. Available: https://ieeexplore.ieee.org/document/4638746

[9] F. SICARD, E. ZAMAI, and J.-M. FLAUS, "Filters based approach with temporal and combinational constraints for cybersecurity of industrial control systems." *10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS*, vol. 51, pp. 96–103, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318322328

[10] O. Gonda, "Understanding the threat to scada networks." *Network Security*, vol. 2014, pp. 17–18, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1353485814700935

[11] D. Aitel, "Cybersecurity essentials for electric operators," *The Electricity Journal*, vol. 26, pp. 52–58, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1040619012003089?via

[12] B. Miller and D. C. R. P. D, "A survey of scada and critical infrastructure incidents," *In Proceedings of the 1st Annual conference on Research in information technology, RIIT 12*, pp. 51–56, 2012. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/2380790.2380805

[13] H. S. Snchez, D. Rotondo, T. Escobeta, V. Puigab, J. Saludes, and J. Quevedoa, "Detection of replay attacks in cyber-physical systems using a frequency-based signature." *Journal of the Franklin Institute*, vol. 356, pp. 2798–2824, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0016003219300134

[14] J. Scott, S. Fellow, and D. Spaniel, "Rise of the machines: The dyn attack was just a practice run," *Institute For Critical Infrastructure Technology*, vol. 62, pp. 16–17, 2016. [Online]. Available: https://icitech.org/wp-content/uploads/2016/12/ICIT-Brief-Rise-of-the-Machines.pdf

[15] C. Bronk and E. Tikk-Ringas, "Rise of the machines: The dyn attack was just a practice run," *SURVIVAL,GLOBAL POLITICS AND STRATEGY*, vol. 30, pp. 4–6, 2013. [Online]. Available: https://www.amazon.com/Rise-Machines-Attack-Just-Practice/dp/1540894576

[16] M. Bozdal, M. Randaa, M. Samiea, and I. Jennionsa, "Hardware trojan enabled denial of service attack on can bus," *Procedia Manufacturing*, vol. 6, pp. 1–2, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978918312794?via%3Dihub

[17] Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, and K. Stoddar, "A review of cyber security risk assessment methods for scada systems." *computers & security*, vol. 56, pp. 1–27, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404815001388

[18] N. Sayfayn and S. Madnick, "Cybersafety analysis of the maroochy shire sewage spill," *Management Sloan School*, vol. 29, 2017. [Online]. Available: http://web.mit.edu/smadnick/www/wp/2017-09.pdf

[19] P. K. Chan, M. V. Mahoney, and M. H. Arshad, "A machine learning approach to anomaly detection," 2003. [Online]. Available: dspace-test.lib.fit.edu/bitstream/handle/11141/114/cs-2003-06.pdf?sequence=1

[20] Wang, Li, Sng, and Dennis, "Deep learning algorithms with applications to video analytics for a smart city: A survey," 12 2015. [Online]. Available: https://arxiv.org/pdf/1512.03131.pdf

[21] J. Patterson and A. Gibson, *Deep Learning*, 1st ed. O'Reilly Media, Inc, 2017. [Online]. Available: https://learning.oreilly.com/library/view/deep-learning/9781491924570/ch01.html

[22] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classication of malware: Research developments, trends and challenges," *Journal of Network and Computer Applications*, 2013. [Online]. Available: sciencedirect.com/science/article/pii/S1084804519303868

[23] T. Xia, Guangzhi, S. Hariri, and M. Yousif, "An efficient network intrusion detection method based on information theory and genetic algorithm," *24th IEEE International Performance, Computing, and Communications Conference*, 2005. [Online]. Available: ieeexplore.ieee.org/abstract/document/1460505

[24] O. Niggemann and V. Lohweg, "On the diagnosis of cyber-physical production systems: State-of-the-art and research agenda," *aaai.org*, vol. 8, 2015. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9530/9691

[25] T. C. Elena Baralis, "Clustering fundamentals," 2019. [Online]. Available: http://dbdmg.polito.it/wordpress/wp-content/uploads/2018/10/9-DMClustering.pdf

[26] C. Elodie and S. Audine, "Diagnosis approaches for detection and isolation of cyber attacks and faults on a two-tank system," *Archive ouverte HAL*, 2013. [Online]. Available: https://hal.laas.fr/hal-02439489/document

[27] B. O. Bouamama, "Modlisation et supervision des systmes en gnie des procds - approche bond graphs," *Archive ouverte HAL*, 2011. [Online]. Available: https://hal.archives-ouvertes.fr/tel-01736792/document

[28] "Math. graphics. programming. (matlab application)," https://www.mathworks.com/products/matlab.html, accessed: 2020-05-19.

[29] "Classification learner app," https://www.mathworks.com/help/stats/classificationlearner-app.html, accessed: 2020-05-19.

[30] "Simulation and modelbased design (simulink app)," https://www.mathworks.com/products/simulink.html, accessed: 2020-05-19.